WHAT IS CLAIMED IS:

1. A method implemented in circuitry, comprising:

accessing a program comprising a plurality of instructions including at least one no operation (NOP) instruction; and

removing at least one NOP instruction in the program that is not needed to provide a processing delay to ensure data is available to at least one dependent instruction accessing the data.

2. The method of claim 1, wherein removing the at least one NOP instruction comprises at least one of:

deleting one NOP instruction in the program that is not needed to provide the processing delay to ensure the data is available to at least one dependent instruction without moving a non-NOP instruction; and

replacing one NOP instruction with one determined instruction preceding the NOP instruction in response to determining that one instruction preceding at least one NOP instruction is capable of replacing the NOP instruction in the program.

3. The method of claim 1, further comprising:

deleting at least one instruction in the program that is not needed to provide the processing delay to ensure the data is available to at least one dependent instruction; and

after deleting the at least one instruction, replacing at least one NOP instruction with one determined instruction preceding the at least one NOP instruction in response to determining that one instruction preceding at least one NOP instruction is capable of replacing the NOP instruction in the program.

4. The method of claim 3, further comprising:

performing an additional iteration of deleting at least one instruction and then replacing the at least one NOP instruction in response to replacing at least one NOP instruction.

5. The method of claim 1, wherein the instructions in the program comprise assembly language instructions coded by a developer.

6. The method of claim 1, wherein removing NOP instructions in the program further comprises accessing and processing each NOP instruction by:

determining whether the accessed NOP instruction is needed to delay processing of one dependent instruction following the accessed NOP instruction to ensure that data is available to the dependent instruction accessing the data; and

deleting the accessed NOP instruction in response to determining that the NOP instruction is not needed to ensure that data is available to the dependent instruction accessing the data..

7. The method of claim 6, wherein determining whether the accessed NOP instruction is needed to delay processing of one dependent instruction further comprises:

identifying instructions preceding the NOP instruction that have a delay in writing the results. and

identifying dependent instructions following the NOP instruction that are dependent on an availability of data from the identified instructions having the delay in writing the results.

8. The method of claim 1, wherein removing NOP instructions in the program further comprises accessing and processing each NOP instruction by:

replacing the accessed NOP instruction with one previous non-NOP instruction that is capable of being moved forward to replace the accessed NOP instruction without preventing data from being available to one dependent instruction following the NOP instruction.

9. The method of claim 8, wherein the one previous instruction comprises a preceding instruction closest to the accessed NOP instruction in the program.

10. The method of claim 8, wherein removing the NOP instructions further comprises:

deleting at least one NOP instruction not needed to ensure that data accessed by the dependent instruction is available to the dependent instruction, wherein the operations of replacing accessed NOP instructions with previous non-NOP instructions are performed after deleting NOP instructions not needed to ensure that data accessed by the dependent instruction is available.

11. The method of claim 1, wherein removing NOP instructions in the program further comprises accessing each NOP instruction and performing:

replacing the accessed NOP instruction with one previous non-NOP instruction that is capable of being moved forward to replace the accessed NOP instruction without preventing data from being available to one dependent instruction following the NOP instruction and that is not a branch target instruction.

12. The method of claim 1, wherein the program instructions are for execution by an engine in a multiprocessor engine.

13. A system for processing a plurality of instructions including at least one no operation (NOP) instruction, comprising:

circuitry operable to:

access the program; and

remove at least one NOP instruction in the program that is not needed to provide a processing delay to ensure data is available to at least one dependent instruction accessing the data.

14. The system of claim 13, wherein the operation to remove the at least one NOP instruction is further operable to:

delete one NOP instruction in the program that is not needed to provide the processing delay to ensure the data is available to at least one dependent instruction without moving a non-NOP instruction; and replace one NOP instruction,with one determined instruction preceding the NOP instruction in response to determining that one instruction preceding at least one NOP instruction is capable of replacing the NOP instruction in the program.

15. The system of claim 13, wherein the circuitry is further operable to:

delete at least one instruction in the program that is not needed to provide the processing delay to ensure the data is available to at least one dependent instruction; and

after deleting the at least one instruction, replace at least one NOP instruction with one determined instruction preceding the at least one NOP instruction in response to determining that one instruction preceding at least one NOP instruction is capable of replacing the NOP instruction in the program.

16. The system of claim 15, wherein the circuitry is further operable to:

perform an additional iteration of deleting at least one instruction and then replacing the at least one NOP instruction in response to replacing at least one NOP instruction.

17. The system of claim 13, wherein the instructions in the program comprise assembly language instructions coded by a developer.

18. The system of claim 13, wherein the operation to remove NOP instructions in the program further comprises accessing and processing each NOP instruction to:

determine whether the accessed NOP instruction is needed to delay processing of one dependent instruction following the accessed NOP instruction to ensure that data is available to the dependent instruction accessing the data; and

delete the accessed NOP instruction in response to determining that the NOP instruction is not needed to ensure that data is available to the dependent instruction accessing the data.

19. The system of claim 18, wherein the operation to determine whether the accessed NOP instruction is needed to delay processing of one dependent instruction is further operable to:

identify instructions preceding the NOP instruction that have a delay in writing the results. and

identify dependent instructions following the NOP instruction that are dependent on an availability of data from the identified instructions having the delay in writing the results.

20. The system of claim 13, wherein the operation to remove NOP instructions in the program further accesses and processes each NOP instruction to:

replace the accessed NOP instruction with one previous non-NOP instruction that is capable of being moved forward to replace the accessed NOP instruction without preventing data from being available to one dependent instruction following the NOP instruction.

21. The system of claim 21, wherein the one previous instruction comprises a preceding instruction closest to the accessed NOP instruction in the program.

22. The system of claim 13, wherein the operation to remove the NOP instructions is further operable to:

delete at least one NOP instruction not needed to ensure that data accessed by the dependent instruction is available to the dependent instruction, wherein the operations of replacing accessed NOP instructions with previous non-NOP instructions are performed after deleting NOP instructions not needed to ensure that data accessed by the dependent instruction is available.

23. The system of claim 13, wherein the operation to remove NOP instructions in the program further accesses each NOP instruction to:

replace the accessed NOP instruction with one previous non-NOP instruction that is capable of being moved forward to replace the accessed NOP instruction without preventing data from being available to one dependent instruction following the NOP instruction and that is not a branch target instruction.

24. An article of manufacture enabled to:

access a program comprising a plurality of instructions including at least one no operation (NOP) instruction; and

remove at least one NOP instruction in the program that is not needed to provide a processing delay to ensure data is available to at least one dependent instruction accessing the data.

25. The article of manufacture of claim 24, wherein the operation to remove the at least one NOP instruction is further operable to:

delete one NOP instruction in the program that is not needed to provide the processing delay to ensure the data is available to at least one dependent instruction without moving a non-NOP instruction; and

replace one NOP instruction with one determined instruction preceding the NOP instruction in response to determining that one instruction preceding at least one NOP instruction is capable of replacing the NOP instruction in the program.

26. The article of manufacture of claim 24, wherein the article of manufacture is further operable to:

delete at least one instruction in the program that is not needed to provide the processing delay to ensure the data is available to at least one dependent instruction; and

after deleting the at least one instruction, replace at least one NOP instruction with one determined instruction preceding the at least one NOP instruction in response to determining that one instruction preceding at least one NOP instruction is capable of replacing the NOP instruction in the program.

27 The article of manufacture of claim 26, wherein the article of manufacture is further operable to:

perform an additional iteration of deleting at least one instruction and then replacing the at least one NOP instruction in response to replacing at least one NOP instruction.

28. The article of manufacture of claim 24, wherein the instructions in the program comprise assembly language instructions coded by a developer.

29. The article of manufacture of claim 24, wherein the operation to remove NOP instructions in the program further accesses and processes each NOP instruction to:

determine whether the accessed NOP instruction is needed to delay processing of one dependent instruction following the accessed NOP instruction to ensure that data is available to the dependent instruction accessing the data; and

delete the accessed NOP instruction in response to determining that the NOP instruction is not needed to ensure that data is available to the dependent instruction accessing the data.

30. The article of manufacture of claim 29, wherein the operation to determine whether the accessed NOP instruction is needed to delay processing of one dependent instruction is further operable to:

identify instructions preceding the NOP instruction that have a delay in writing the results. and

identify dependent instructions following the NOP instruction that are dependent on a availability of data from the identified instructions having the delay in writing the results.

31. The article of manufacture of claim 24, wherein the operation to remove NOP instructions in the program further accesses and processes each NOP instruction to:

replace the accessed NOP instruction with one previous non-NOP instruction that is capable of being moved forward to replace the accessed NOP instruction without preventing data from being available to one dependent instruction following the NOP instruction.

32. The article of manufacture of claim 31, wherein the one previous instruction comprises a preceding instruction closest to the accessed NOP instruction in the program.

33. The article of manufacture of claim 31, wherein the operation to remove the NOP instructions is further operable to:

delete at least one NOP instruction not needed to ensure that data accessed by the dependent instruction is available to the dependent instruction, wherein the operations of replacing accessed NOP instructions with previous non-NOP instructions are performed after deleting NOP instructions not needed to ensure that data accessed by the dependent instruction is available.

34. The article of manufacture of claim 24, wherein the operation to remove NOP instructions in the program further accesses each NOP instruction to:

replace the accessed NOP instruction with one previous non-NOP instruction that is capable of being moved forward to replace the accessed NOP instruction without preventing data from being available to one dependent instruction following the NOP instruction and that is not a branch target instruction.

35. The article of manufacture of claim 24, wherein the program instructions are for execution by an engine in a multiprocessor engine.